Proper RFID Privacy: Model and Protocols

Jens Hermans, Roel Peeters, and Bart Preneel
KU Leuven ESAT/COSIC & iMinds, Belgium
Email: firstname.lastname@esat.kuleuven.be

***Abstract*—We approach RFID privacy both from modelling and protocol point of view. Our privacy model avoids the drawbacks of several proposed RFID privacy models that either suffer from insufficient generality or put forward unrealistic assumptions regarding the adversary's ability to corrupt tags. Furthermore, our model can handle multiple readers and introduces two new privacy notions to capture the recently discovered insider attackers.**

We analyse multiple existing RFID protocols, demonstrating the easy applicability of our model, and propose a new wide-forward-insider private RFID authentication protocol. This protocol provides sufficient privacy guarantees for most practical applications and is the most efficient of its kind, it only requires two scalar-EC point multiplications.

***Index Terms*—Computer security, authentication, privacy, cryptography, RFID tags.**

I. Introduction

As Radio Frequency Identification (RFID) systems are becoming more common (for example in access control, product tracing, e-ticketing, electronic passports), managing the associated privacy and security concerns becomes more important [1]. Since RFID tags are primarily used for authentication purposes, security in this context means that it should be infeasible to impersonate a legitimate tag. Privacy, on the other hand, means that unauthorized parties should not be able to identify, trace, or link tag appearances.

For RFID tags used in consumer applications, an adversary typically learns the outcome of the identification protocol, on top of sending arbitrary queries to tags and getting responses. Successful identifications result in an unlocked door, unlocked car or processed payment; while failure has no outcome. As low-cost devices, RFID tags are hardly protected against physical tampering. In particular, it has been shown that side-channel attacks may enable an adversary to extract secrets from the tag [2], [3], [4]. Furthermore, by inducing power drops or by otherwise influencing the physical environment of the tag, so-called 'reset' attacks enable the adversary to force the tag to re-use old randomness [5], [6], [7].

Several models for privacy and security in the context of RFID systems have been proposed in the literature. Section III discusses a selection of *general* models. For some of these models we show that, despite their intended generality, it remains unclear how to apply these to protocols other than the protocol in the context of which they were proposed. Other existing models do not allow for adversaries that corrupt tags.

So far, little attention has been paid to supporting multiple readers. Most RFID models only take into account the limited setting of one reader. Multiple readers occur for example when using a single RFID card to access multiple disjoint security systems (e.g. multiple buildings, printer systems, vending machines). Supporting multiple readers, however severely complicates the setup since it is more likely that one of the readers will be compromised. This should not affect the security of the tags authenticating to other uncompromised readers. As such, sharing secrets among readers is impossible.

We present a new RFID privacy model in Sect. IV that is robust in the sense that it can handle tampering with tags, is easily applicable, general and can handle multiple readers. Furthermore, we introduce two new privacy notions to also cover the case where an adversary uses the internal state from a corrupted tag to attack privacy of other tags. These 'insider' attacks were described by van Deursen and Radomirović [8], clearly showing that wide-forward[1] private protocols are not sufficient. At first glance wide-forward privacy seems to imply that even when the outcome of protocol is known to the adversary, all interactions of a tag up to the point of corruption cannot be linked. However, it was shown for two wide-forward private protocols, that the adversary can link uncorrupted tags, when given the ability to learn the outcome of the protocol and the state of one legitimate 'insider' tag.

Using this new model as a reference we evaluate the privacy of several existing RFID protocols in Sect. V and we design and evaluate a new wide-forward-insider private RFID identification protocol in Sect. VI. An optimised version of this protocol is discussed in Sect. VII. After listing some implementation considerations for private RFID authentication protocols in Sect. VIII, we compare the security, privacy and performance features of our protocol with the discussed previously proposed protocols. This not only validates our new model, but also shows that our new optimised protocol is the most efficient one, only requiring two scalar-EC point multiplications.

II. Definitions

A. RFID System

Throughout this article we use a common model for RFID systems, similar to the definitions introduced in [9], [10]. For our model, we extend these definitions to allow for multiple readers, unlike previously proposed models.

Definition 1 (RFID System)**.** *An RFID system consists of a set of tags $\mathcal{T}$ and a set of readers $\mathcal{R}$. Each tag is identified by an identifier ID. The memory of the tags contains a state S, which may change during the lifetime of the tag. The tag's ID may or may not be stored in S. Each tag is a transponder with limited memory and computation capability. A reader R_j consists of*

[1] An overview of the different privacy notions is given in Sect. III

one or more transceivers and a database. A reader's task is to identify legitimate tags (i.e. to recover their IDs), and to reject all other incoming communication. Each reader has a database that contains for every tag T_i, its ID and a matching secret K_{R_j,T_i}. The secret K_{R_j,T_i} is specific for every tag and can be different for every tag-reader combination. A tag is 'registered' with a reader if the reader database contains an entry for that tag and can successfully authenticate it.

In general, an RFID system requires several algorithms and protocols for setting up the readers, tags, registering tags with readers or even unregistrering tags. These routines are highly dependent on the specific layout of the RFID system and can even take place offline. The privacy and security of these setup and registration algorithms and protocols is thus outside of the scope of this article. During the remainder of the article we will only discuss the main protocol used for tag identification and none of the auxiliary setup and registration routines.

B. Notation

If $\mathcal{T}$ is a set, $t \in_R \mathcal{T}$ means that t is chosen uniformly at random from $\mathcal{T}$. $|\mathcal{T}|$ denotes the cardinality of the set. If $\mathcal{A}$ is an algorithm and $\mathcal{O}$ an oracle, then $\mathcal{A}^{\mathcal{O}}$ denotes the fact that $\mathcal{A}$ has access to the oracle $\mathcal{O}$.

Our proposed protocol is based on Elliptic Curve Cryptography, hence we make use of additive notation. Points on the curve are represented by capital letters while scalars are represented by lowercase letters.

The $\texttt{xcoord}(\cdot)$ function comes almost for free when using elliptic curves. Assuming an elliptic curve $\mathbb{E}$ with prime order ℓ over $\mathbb{F}_p$, then for a point $Q = \{q_x, q_y\}$ with $q_x, q_y \in [0 \dots p-1]$, $\texttt{xcoord}(Q)$ maps Q to $q_x \mod \ell$.

A function $f : \mathbb{N} \to \mathbb{R}$ is called 'polynomial' in the security parameter $k \in \mathbb{N}$ if $f(k) = O(k^n)$, with $n \in \mathbb{N}$. It is called 'negligible' if, for every $c \in \mathbb{N}$ there exists an integer k_c such that $f(k) \leq k^{-c}$ for all $k > k_c$. We denote a negligible function by ϵ.

C. Number-theoretical Assumptions

1) Discrete Logarithm: Let P be a generator of a group $\mathbb{G}_\ell$ of order ℓ and let A be a given arbitrary element of $\mathbb{G}_\ell$. The discrete logarithm (DL) problem is to find the unique integer $a \in \mathbb{Z}_\ell$ such that $A = aP$. The DL assumption states that it is computationally hard to solve the DL problem.

2) One More Discrete Logarithm: The one more discrete logarithm (OMDL) problem was introduced by Bellare [11]. Let P be a generator of a group $\mathbb{G}_\ell$ of order ℓ. Let $\mathcal{O}_1()$ be an oracle that returns random elements $A_i = a_i P$ of $\mathbb{G}_\ell$. Let $\mathcal{O}_2(\cdot)$ be an oracle that returns the discrete logarithm of a given input base P. The OMDL problem is to return the discrete logarithms for each of the elements obtained from the m queries to $\mathcal{O}_1()$, while making strictly less than m queries to $\mathcal{O}_2(\cdot)$ (with $m > 0$).

3) x-Logarithm: Brown and Gjøsteen [12] introduced the x-Logarithm (XL) problem: given an elliptic curve point, determine whether its discrete logarithm is congruent to the x-coordinate of an elliptic curve point. The XL assumption states that it is computationally hard to solve the XL problem. Brown and Gjøsteen also provided some evidence that the XL problem is almost as hard as the DDH problem.

4) Diffie-Hellman: Let P be a generator of a group $\mathbb{G}_\ell$ of order ℓ and let aP, bP be two given arbitrary elements of $\mathbb{G}_\ell$, with $a, b \in \mathbb{Z}_\ell$. The computational Diffie-Hellman (CDH) problem is, given P, aP and bP, to find abP. The 4-tuple $\langle P, aP, bP, abP\rangle$ is called a Diffie-Hellman tuple. Given a fourth element $cP \in \mathbb{G}_\ell$, the decisional Diffie-Hellman (DDH) problem is to determine if $\langle P, aP, bP, cP\rangle$ is a valid Diffie-Hellman tuple or not.

5) Oracle Diffie-Hellman: Abdalla *et al.* [13] introduced the ODH assumption:

Definition 2. *Oracle Diffie-Hellman (ODH) Given $A = aP, B = bP$, a function H and an adversary $\mathcal{A}$, consider the following experiments:*

Experiment $\mathbf{Exp}^{odh}_{H,\mathcal{A}}$:

- $\mathcal{O}(Z) := H(bZ)$ *for* $Z \neq \pm A$
- $g = \mathcal{A}^{\mathcal{O}(\cdot)}(A, B, H(C))$
- *Return* g

The value C is equal to abP for the $\mathbf{Exp}^{odh-real}_{H,\mathcal{A}}$ experiment, chosen at random in $\mathbb{G}_\ell$ for the $\mathbf{Exp}^{odh-random}_{H,\mathcal{A}}$ experiment.

We define the advantage of $\mathcal{A}$ violating the ODH assumption as:

$$|\Pr\left[\mathbf{Exp}^{odh-real}_{H,\mathcal{A}} = 1\right] - \Pr\left[\mathbf{Exp}^{odh-random}_{H,\mathcal{A}} = 1\right]|.$$

The ODH assumption consists of the plain DDH assumption combined with an additional assumption on the function $H(\cdot)$. The idea is to give the adversary access to an oracle $\mathcal{O}$ that computes bZ, without giving the adversary the ability to compute bA, which can then be compared with C. To achieve this one restricts the oracle to $Z \neq \pm A$, and moreover, only $H(bZ)$ instead of bZ is released, to prevent the adversary from exploiting the self-reducibility of the DL problem.[2] The crucial property that is required for $H(\cdot)$ is one-wayness.

III. Existing Privacy Models

This section discusses certain existing RFID privacy models. These models usually consist of a correctness (no false negatives), security (no false positives) and privacy definition.

Note that covering all existing models would exceed the scope of this article by far. Many models, including the ones introduced in [14], [15], [16], [17], [18], [19] do not allow corrupted tags to be traced. We have selected one such model, *i.e.* Juels-Weis [19], for further discussion, in addition to the stronger models of Vaudenay [9] and Canard *et al.* [10].

A. Vaudenay

Several concepts from the privacy model introduced by Vaudenay [9] are used in our model. Therefore, we present this model in detail.

[2]The adversary can set $Z = rA$ for known r and compute $r^{-1}(bZ) = bA$.

1) Adversarial model: The adversary has the ability to influence all communication between a tag and the reader and can therefore perform man-in-the-middle attacks on any tag within its range. It may also obtain the result of the authentication of a tag, *i.e.* whether the reader accepts or rejects the tag. The adversary may also 'draw' (at random) tags and then 'free' them again, moving them inside and outside its range. During these interactions the adversary is given a virtual reference $vtag$ (not the tag's real reference T_i) in order to refer to the tags that are inside its range. Finally the adversary may corrupt tags, thereby learning their entire internal state.

The above interactions take place over eight oracles that the adversary may invoke: `CreateTag(ID)`, `DrawTag(distr)` $\rightarrow (vtag)$, `Free(`$vtag$`)`, `Launch` $\rightarrow \pi$, `SendReader(`m, π`)` $\rightarrow m'$, `SendTag(`$m, vtag$`)` $\rightarrow m'$, `Result(`π`)` $\rightarrow x$ and `Corrupt(`$vtag$`)`. Let $vtag$ denote a virtual tag reference, π a protocol instance, $distr$ a polynomially bounded sampling algorithm, m and m' messages and ID the tag's identifier. For a complete definition of the oracles the reader is referred to [9].

The Vaudenay model divides adversaries into different classes, depending on restrictions regarding their use of the above the oracles. In particular, a *strong* adversary may use all eight oracles without any restrictions. A *destructive* adversary is not allowed to interact with a tag after it has been corrupted. This models situations where corrupting a tag leads to the destruction of the tag. A *forward* adversary can only do other corruptions after the first corruption. That is, no protocol interactions are allowed after the first call to the `Corrupt` oracle. A *weak* adversary does not have the ability to corrupt tags. Orthogonal to these four attacker classes there is the notion of *wide* and *narrow* adversary. A *wide* adversary has access to the result of the verification by the server while a *narrow* adversary does not. The most important relations between the above privacy notions are given below:

wide-strong	$\Rightarrow$	wide-destructive	$\Rightarrow$	wide-forward	$\Rightarrow$	wide-weak
$\Downarrow$		$\Downarrow$		$\Downarrow$		$\Downarrow$
narrow-strong	$\Rightarrow$	narrow-destructive	$\Rightarrow$	narrow-forward	$\Rightarrow$	narrow-weak

In this case $A \Rightarrow B$ means that if the protocol is A-private it implies that the protocol is B-private. A protocol that is *wide-strong* private, for example, obviously also belongs to all other privacy classes, that only allow weaker adversaries.

Due to their generality, the above restrictions can be used perfectly in other privacy models. Throughout the article we will frequently refer to strong, destructive, forward, weak and wide/narrow adversaries.

2) Privacy, security and correctness: Privacy is defined by means of the notion of a 'trivial' adversary. Intuitively, a trivial adversary does not 'use' the communication captured during the protocol run to determine its output.

Definition 3 (Blinder, trivial adversary - <Simplified> Def. 7 from [9])**.** *A Blinder $\mathcal{B}$ for an adversary $\mathcal{A}$ is a polynomial-time algorithm which sees the messages that $\mathcal{A}$ sends and receives, and simulates the* `Launch`*,* `SendReader`*,* `SendTag` *and* `Result` *oracles to $\mathcal{A}$. The blinder does not have access to the reader tapes. A blinded adversary $\mathcal{A}^{\mathcal{B}}$ is an adversary who does not use the* `Launch`*,* `SendReader`*,* `SendTag` *and* `Result` *oracles.*

An adversary $\mathcal{A}$ is trivial if there exists a blinder $\mathcal{B}$ such that $|\Pr(\mathcal{A}\,wins) - \Pr(\mathcal{A}^{\mathcal{B}}\,wins)|$ is negligible.

Intuitively, an adversary is called trivial if, even when blinded, it still produces the same output. Such an adversary does not 'use' the communication captured during the protocol run in order to determine its output. Note that a blinded adversary is not the same as a simulator typically found in security proofs: the blinder is separate from the adversary and has no access to the adversary's tape. The blinder just receives incoming queries from the adversary and has to respond either by itself or by forwarding the queries to the system. We are now ready to present the privacy definition.

Definition 4 (Privacy - <Simplified> Def. 6 from [9])**.** *The privacy game between the challenger and the adversary consists of two phases:*

1) *Attack phase: the adversary issues oracle queries according to applicable restrictions.*
2) *Analysis phase: the adversary receives the table that maps every $vtag$ to a real tag ID. Then it outputs* `true` *or* `false`*.*

The adversary wins if it outputs `true`*. A protocol is called X-private, where X is an adversary class (strong, destructive, ...), if and only if all winning adversaries that belong to the class X are trivial.*

Besides privacy the protocol should also offer authentication of the tag. We refer to this property as the *security* of the protocol.

Definition 5 (Security - <Simplified> Def. 4 from [9])**.** *We consider any adversary in the class* strong*. The adversary wins if the reader identifies an uncorrupted legitimate tag, but the tag and the reader did not have a matching conversation. The RFID scheme is called secure if the success probability of any such adversary is negligible.*

Definition 6 (Correctness - Def. 1 from [9])**.** *An RFID scheme is correct if its output is correct except with negligible probability for any polynomial-time experiment which can be described as follows:*

1) *Set up the reader.*
2) *Create a number of tags including one named ID.*
3) *Execute a complete protocol between reader and tag ID.*

The output is correct if and only if $Output = \perp$ and tag ID is not legitimate or $Output = ID$ and tag ID is legitimate.

In a follow-up paper by Paise and Vaudenay [20], the concept of mutual authentication for RFID is defined. The tag simply outputs a boolean, indicating whether or not the reader was accepted. The authors extend the security definition by adding a criterion for reader authentication.

3) Discussion: The paper of Vaudenay inspired many authors to formulate derived RFID privacy models or to evaluate the (Paise-)Vaudenay model [10], [20], [21], [22], [23], [24]. Although Vaudenay's privacy model is perhaps the strongest and most complete, it contains some flaws with respect to strong privacy.

Vaudenay's proof of the statement that "wide-strong privacy is impossible" uncovers some of these flaws. This proof assumes a wide-destructive private protocol. By definition, for every destructive adversary, there exists a blinder. This includes the adversary that (a) creates one real tag, (b) corrupts this tag right away, (c) starts a protocol using either the state from the corrupted tag or from another fake tag. In the end, the blinder has to answer the `Result` oracle. Obviously, the adversary knows which tag was selected and knows which result to expect. However, since the blinder has no access to this random coin of the adversary, it must be able to distinguish a real and a fake tag just by looking at the protocol run from the side of the reader. The proof then uses this blinder to construct a wide-strong adversary. Since all wide-strong adversaries are also wide-destructive, this proves the impossibility of wide-strong privacy.

Obviously, this proof only works because the blinder is separated from the adversary. The issue with a separate blinder is exploited multiple times by Armknecht *et al.* [25] in the Paise-Vaudenay [20] model. Using this property the authors show the impossibility of reader authentication combined with respectively narrow forward privacy (if `Corrupt` reveals the temporary state of tags) and narrow strong privacy (if `Corrupt` only reveals the permanent state of tags). In later work [26], Ouafi and Vaudenay correct the inconsistency in the model and shows that strong privacy is indeed possible. In this new approach, the blinder is given access to the random coin flips of the adversary.

Independent from this correction, Ng *et al.* [21] also identified the problems with strong privacy. They propose a solution, based on the concept of a 'wise' adversary that does not make any 'irrelevant' queries to the oracles i.e. queries to which it already knows the answer. The authors claim that, if the protocol does not generate false negatives, then a wise adversary never calls the `Result` oracle. Given the vague definition of wise adversaries it is hard to verify these claims. The existence of attacks which exploit false positives [27] however, suggests that the general claim that `Result` is not used by a wise adversary is incorrect. Based on this questionable general claim, the authors further identify an IND-CPA-based protocol as being strong private, without giving a formal proof.[3]

B. Canard et al.

1) Model: The model of Canard *et al.* [10] builds on the work of Vaudenay, so the definition of oracles is quite similar. For the privacy definition the model requires the adversary to produce a non-obvious link between virtual tags.

Definition 7. *$(vtag_i, vtag_j)$ is a non-obvious link if $vtag_i$ and $vtag_j$ refer to the same ID and if a 'dummy' adversary, who only has access to `CreateTag`, `Draw`, `Free`, `Corrupt`, is not able to output this link with a probability better than $1/2$.*

The definition requires the adversary to output a non-obvious link to win the privacy game. A protocol is said to be untraceable if, for every adversary $\mathcal{A}$, it is possible to construct a 'dummy' adversary $\mathcal{A}_d$ such that $|\mathbf{Succ}_{\mathcal{A}}^{Unt}(1^k) - \mathbf{Succ}_{\mathcal{A}_d}^{Unt}(1^k)| \leq \epsilon(k)$.

It is unclear why the authors use the probability threshold $1/2$, since one would expect some dependency on the total number of non-obvious links.[4]

One major difference with respect to Vaudenay's model is that a 'dummy' adversary is used instead of a blinded adversary. This avoids some of the issues surrounding the use of a blinder, because a 'dummy' adversary can also access its own random tape, while a blinder cannot access the adversary's random tape.

2) Discussion: While the work certainly has its merit in formalizing and fixing the Vaudenay model (by using a dummy adversary instead of a blinder), the model of Canard *et al.* lacks generality because it focuses on non-trivial links. Moreover the above definitions are circular: the definition of a non-obvious link refers to a dummy adversary, which on its turn is defined in terms of the probability of outputting a non-obvious link. In its current form, without changing any definition to break the circularity, the model is not useable.

C. Juels-Weis

1) Model: The model by Juels and Weis [19] is based on the notion of indistinguishability. The model does not feature a `DrawTag` query and the `Corrupt` query is replaced by a `SetKey` query, which returns the current secret of the tag and allows the adversary to set a new secret. Figure 1 shows a simplified version of the privacy game. The protocol is considered private if $\forall \mathcal{A}$, $\Pr\left[\mathbf{Exp}_{\mathcal{A},\mathcal{S}}^{priv} \text{guesses } b \text{ correctly}\right] \leq \frac{1}{2} + \epsilon$.

Experiment $\mathbf{Exp}_{\mathcal{A},\mathcal{S}}^{priv}$:

1) Setup:
   - Generate n random keys key_i.
   - Initialize the reader with the random key_i.
   - Create n tags, each with a key_i.
2) Phase (1): Learning
   - $\mathcal{A}$ can interact with a polynomial number of calls to the system, but can only issue `SetKey` on $n-2$ tags, leaving at least 2 uncorrupted tags
3) Phase (2): Challenge
   - $\mathcal{A}$ selects two uncorrupted tags T_0 and T_1. Both are removed from the set of tags.
   - One of these tags (T_b, the challenge tag) will be selected at random by the challenger.
   - $\mathcal{A}$ can make a polynomial number of calls to the system, but cannot corrupt the challenge tag T_b.
   - $\mathcal{A}$ outputs a guess bit $g \in \{0, 1\}$.

Figure 1. Privacy experiment from Juels-Weis

2) Discussion: The Juels-Weis model is one of the few models that are based on a indistinguishability game instead of the notion of simulatability. The model is limited by the fact that the challenge tags cannot be corrupted. In terms of the Vaudenay model [9] it would be a weak adversary with regard to the challenge tags. For example, attacks in which the adversary links together executions of a tag that have taken place prior to its corruption are not possible in the Juels-Weis model because of this.

[3] Note that the original security proof (*i.e.* no false positives) by Vaudenay requires IND-CCA2 encryption, so using only IND-CPA encryption would require a new security proof. `Result` may serve as a decryption oracle.

[4] One slightly different interpretation is that a 'dummy' adversary cannot determine if a given non-obvious candidate link $vtag_i$, $vtag_j$ is a link in reality or not. This interpretation however contradicts the definition of the success probability of an adversary given in the paper.

The model from Ha *et al.* [18] is very similar, with the difference that the privacy is defined as distinguishing the reply from a real tag from a random reply.

D. Bohli-Pashalidis

1) Model: Unlike the previous models, the Bohli-Pashalidis[28] model is not an RFID-specific model. Unfortunately, it captures only privacy properties; properties like security and correctness are not covered. The model considers a set of users (with unique identifiers) $\mathcal{U}$, whose size is at least polynomial in a security parameter. There is no formal difference between different types of players, like there is with tag and reader in most RFID models. The system $\mathcal{S}$ can be invoked with input batches $(u_1, \alpha_1), (u_2, \alpha_2), \ldots, (u_c, \alpha_c) \in (\mathcal{U}, A)^c$, consisting of pairs of user identifiers and 'parameters' and will output a batch $((e_1, \ldots e_c), \beta)$, with the outputs e_i from each system invocation and a general output β, applying to the batch as a whole. Users can also be corrupted, revealing their internal state to the adversary.

The authors investigate the properties of the function $f \in \mathcal{F}$, where $\mathcal{F} = \{f : \{1, 2, \ldots, n\} \rightarrow \mathcal{U}\}$ is the space of functions that map the serial number of each output element to the user it corresponds to. In the Strong Anonymity (SA) setting, no information should be revealed to the adversary about the function f, guaranteeing the highest level of privacy. Several weaker notions (which reveal *some* information on f) are defined and the relations among notions are examined.

The adversarial model is based on indistinguishability. The adversary can cause different users to invoke the system using different parameters (e.g. messages) in both a left and right world with the `Input`$((u_0, \alpha_0), (u_1, \alpha_1))$ oracle. Based on a bit b, selected by the challenger, the system will be invoked with the user-data pair (u_b, α_b). That is, the adversary itself defines the functions $f_0, f_1 \in \mathcal{F}$, for respectively the left and right world. The adversary can also corrupt users. At the end of the game the adversary has to output a guess bit g. The adversary wins the game if $g = b$. By imposing restrictions on f_0 and f_1, the authors investigate different levels of privacy.

Definition 8. *A privacy protecting system $\mathcal{S}$ is said to unconditionally provide privacy notion X, if and only if the adversary $\mathcal{A}$ is restricted to invocations (u_0, α_0) and (u_1, α_1) such that f_0 and f_1 are X-indistinguishable for all invocations and for all such adversaries $\mathcal{A}$, it holds that $\mathbf{Adv}^X_{\mathcal{S},\mathcal{A}}(k) = 0$.*

2) Discussion: Due to its generality, and due to the fact that it is not meant to cover security properties, the Bohli-Pashalidis model needs non-trivial adaptations in order to apply to RFID setting. In its current form, the model does not support multi-pass protocols, where linking two messages from the same protocol run is not a privacy breach. Moreover there is no distinction between tags that need to be protected, and the reader for which privacy is not an issue. An interesting question is whether the strictly binary distinguishing game (only one bit of randomness in the challenge) provides enough flexibility compared to other models, like Vaudenay's, where there are multiple bits of randomness that are to be guessed.

IV. Our Model

A. Adversarial Model

We assume a set of readers $\mathcal{R} = \{R_1, R_2, \ldots, R_j\}$ and a set of tags $\mathcal{T} = \{T_1, T_2, \ldots, T_i\}$. $\mathcal{R}$ and $\mathcal{T}$ are initially empty, and readers and tags are added dynamically by the adversary. Each reader maintains a database of tuples (ID_i, K_i), one for every tag $T_i \in \mathcal{T}$ that is registered with that reader. Moreover, every tag T_i stores an internal state S_i.

Let $\mathcal{A}$ denote the adversary, which can adaptively control the system $\mathcal{S}$. $\mathcal{A}$ interacts with $\mathcal{S}$ through a set of oracles. The experiment that the challenger sets up for $\mathcal{A}$ (after the security parameter k is fixed) proceeds as follows:

$\mathbf{Exp}_{\mathcal{S},\mathcal{A}}(k)$:

1) $b \in_R \{0, 1\}$
2) $g \leftarrow \mathcal{A}^{\mathcal{O}}()$
3) Return $g == b$.

At the beginning of the experiment, the challenger picks a random bit b. The adversary $\mathcal{A}$ subsequently interacts with the challenger by means of the set of oracles $\mathcal{O}$:

- `CreateReader()` $\rightarrow R_j$: this oracle creates a new reader. A reference R_j to the new reader is returned.
- `CreateTag(ID)` $\rightarrow T_i$: on input a tag identifier ID, this oracle calls $SetupTag(ID)$ and registers the new tag with the server. A reference T_i to the new tag is returned. Note that this does not reject duplicate IDs.
- `RegisterTag(`T_i`,`R_j`)`: register the tag T_i with the server R_j. The registration of the tag with the reader can be done in several ways (e.g. using a specific protocol that involves both the tag and reader, between readers or using some offline process).
- `Launch(`R_j`)` $\rightarrow \pi$: a new protocol run is launched on the reader R_j, according to the protocol specification. It returns a session identifier π, generated by the reader.
- `DrawTag(`T_i`,`T_j`)` $\rightarrow vtag$: on input a pair of tag references, this oracle generates a virtual tag reference, as a monotonic counter, $vtag$ and stores the triple $(vtag, T_i, T_j)$ in a table $\mathcal{D}$. Depending on the value of b, $vtag$ either refers to T_i or T_j. If one of the two tags T_i or T_j is in the list of insider tags $\mathcal{I}$, $\perp$ is returned and no entry is added to $\mathcal{D}$. If T_i is registered with a different set of readers than T_j, $\perp$ is returned. If T_i is already references as the left-side tag in $\mathcal{D}$ or T_j as the right-side tag, then this oracle also returns $\perp$ and adds no entry to $\mathcal{D}$. Otherwise, it returns $vtag$.
- `Free(vtag)`$_b$: on input $vtag$, this oracle retrieves the triple $(vtag, T_i, T_j)$ from the table $\mathcal{D}$. It resets either T_i (if $b = 0$) or T_j (if $b = 1$). Then it removes the entry $(vtag, T_i, T_j)$ from $\mathcal{D}$. When a tag is reset, its volatile memory is erased. The non-volatile memory, which contains the state S, is preserved.
- `SendTag(vtag,`m`)`$_b \rightarrow m'$: on input $vtag$, this oracle retrieves the triple $(vtag, T_i, T_j)$ from the table $\mathcal{D}$ and sends the message m to either T_i (if $b = 0$) or T_j (if $b = 1$). It returns the reply from the tag (m'). If the above triple is not found in $\mathcal{D}$, it returns $\perp$.
- `SendReader(`R_j`,` π`,` m`)` $\rightarrow m'$: on input π, m this oracle sends the message m to the reader R_j in session π

and returns the reply m' from the reader (if any) is returned by the oracle. [5]

- `Result(`R_j`,`π`)`: on input π, this oracle returns a bit indicating whether or not the reader accepted session π as a protocol run that resulted in successful authentication of a tag. If the session with identifier π is not yet finished, or there exists no session with identifier π, $\perp$ is returned.
- `Corrupt(`T_i`)`: on input a tag reference T_i, this oracle returns the complete internal state of T_i. Note that the adversary is not given control over T_i.
- `CreateInsider(ID)` $\rightarrow T_i, S$: create an insider tag T_i. This runs `CreateTag` to create a new tag T_i and `Corrupt` on the newly created tag. The tag T_i is added to the list $\mathcal{I}$ of insider tags.
- `CorruptReader(`R_j`)`: corrupt the reader R_j, which returns the full database of the reader and all secrets of the reader. Note that in the default privacy game this oracle is not used.

According to the above experiment description, the challenger presents to the adversary the system where either the 'left' tags T_i (if $b = 0$) or the 'right' tags T_j (if $b = 1$) are selected when returning a virtual tag reference in `DrawTag`.

$\mathcal{A}$ queries the oracles a number of times and, subsequently, outputs a guess bit g. We say that $\mathcal{A}$ wins the privacy game if and only if $g = b$, i.e. if it correctly identifies which of the worlds was active. The advantage of the adversary $\mathbf{Adv}_{\mathcal{S},\mathcal{A}}(k)$ is defined as

$$\left| Pr\left[\mathbf{Exp}^0_{\mathcal{S},\mathcal{A}}(k) = 1\right] + Pr\left[\mathbf{Exp}^1_{\mathcal{S},\mathcal{A}}(k) = 1\right] - 1 \right| .$$

Restrictions on tag corruption: In the current setup `Corrupt(`T_i`)` reveals the full internal state of the tag, i.e. both its volatile and non-volatile parts. This follows [25] where it is shown that, if corruptions reveal the volatile state, then the resulting privacy notions are stronger. Single-pass protocols (e.g. challenge-response) do not suffer from any issues, since the volatile memory is typically erased after sending the reply, and hence all computations are confined to the invocation of the `SendTag` oracle. Multi-pass protocols on the contrary, typically require storage of data in between `SendTag` invocations. In this case, corrupting a tag in between protocol passes always reveals the activity of the tag and thus allows trivial attacks on the privacy of the tag. In this case, additional restrictions in the privacy model are required to achieve a reasonable privacy definition for such protocols. Note that mutual authentication always requires a multi-pass protocol.

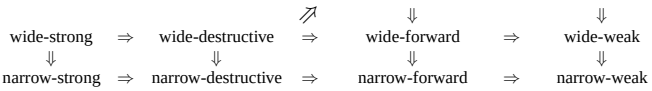
We assume that corrupting a tag implies some kind of physical access to the tag. Such physical access would almost certainly be disruptive for any active protocols being executed on the tag at the time of corruption (if not, one would also need to consider white-box or leakage-resilient cryptography).

Clearly corruption on itself should not yield any advantage to an adversary. For instance, an adversary is could randomly draw one out of two tags and execute several corruptions in between protocol runs while the tag is drawn to see if the selected tag is active. Such an adversary behaviour would also contradict the assumption that physical access is required: drawing a tag at random models the observation of tags passing in proximity of the adversary. The moment an adversary corrupts a tag it breaches the (possible) privacy of such a tag and can certainly identify the tag based on physical properties of the tag or the attached item. Hence it makes no sence to allow corruption of a drawn tag since an adversary should directly select such a tag instead of drawing it. For the remainder of this article we will therefore restrict the `Corrupt` oracle to inactive tags, i.e. tags that are drawn in neither the left or right world.

[5] If no active session π exists, the reader is likely to return $\perp$.

B. Privacy

The adversary restrictions, as defined in Sect. III-A, also apply to our privacy definition. Depending on the acceptable usage of the `Corrupt` oracle, an adversary in our model is either strong, destructive (`Corrupt` destroys a tag), forward (after the first `Corrupt` only further corruptions are allowed), or weak (no `Corrupt` oracle) adversaries. Depending on the allowed usage of the `Result` oracle, there exist narrow (no `Result` oracle) and wide adversaries. If an adversary is allowed to call `CreateInsider` the privacy notion is called 'insider', so we can speak of forward-insider and weak-insider adversaries. For strong and destructive the `CreateInsider` can be simulated using the normal `CreateTag` and `Corrupt` oracles, i.e. strong-insider and destructive-insider are equivalent to strong and destructive respectively. An overview of the relations between the privacy notions, including our two new notions, is given below:



For most practical applications, wide-forward-insider privacy is sufficient. By contrast, the weaker notion of wide-forward privacy is *not* sufficient; indeed, in *e.g.*, transportation systems an adversary has easy access to an insider tag and can thus abuse any privacy guarantees of the system. Furthermore, it seems that the wide-strong notion captures a scenario exceeding the practical requirements for privacy, where an adversary may first corrupt a tag and then release it again for future tracking. In practice this can be done more easily, without physically tampering with the tag itself (*i.e.*, corrupting it), *e.g.*, by attaching a tracking device to the tag.

We now present our definition of privacy. X is used to denote one of the above privacy notions. For the remainder of this article, we will only consider computational privacy.

Definition 9 (Privacy)**.** *An RFID system $\mathcal{S}$, is said to unconditionally provide privacy notion X, if and only if for all adversaries $\mathcal{A}$ of type X, it holds that $\mathbf{Adv}^X_{\mathcal{S},\mathcal{A}}(k) = 0$. Similarly, we speak of computational privacy if for all polynomial time adversaries, $\mathbf{Adv}^X_{\mathcal{S},\mathcal{A}}(k) \leq \epsilon(k)$.*

C. Stateful protocols

Note that stateful protocols (which update their state after a protocol run) do not satisfy our privacy definition. By issuing a `Corrupt(`T_i`)` query before and after a protocol run, one can always identify whether or not the tag has been active.

The main solution we propose is to perform a (possibly random) number of protocol executions on both tags T_i and T_j when `Free` is called. This models the protocol executions that happen outside of the adversaries range, i.e. when a tag is free. These executions are crucial to re-randomize the state of the tag such that consecutive corruption of the tag does not reveal any information.

As an alternative we also mention the X^+ privacy notions, as defined in [28] which can be combined with the first restriction from above. Simply put, this restriction implies that, whenever a tag is corrupted at some point during the privacy game, it always has to be drawn simultaniously in both the left and the right world using a $\texttt{DrawTag}(T_i, T_i)$ query with identical arguments.

D. Correctness and soundness

Correctness ensures that a legitimate tag is always accepted by a reader. Soundness is the property that an illegitimate tag is not accepted by the server. Specifically for the RFID setting, the RFID tag can only participate in one session at the time, hence concurrent attacks on the tag are impossible. This allows us to relax the security definition from requiring matching conversations to impersonation resistance as in [23]. The adversary cannot interact with the tag they try to impersonate during the protocol run between adversary and reader.

Definition 10. *Correctness. A scheme is correct if the identification of a legitimate tag only fails with negligible probability.*

Definition 11. *Soundness. A scheme is resistant against impersonation attacks if no polynomially bounded strong adversary succeeds, with non-negligible probability, in being identified by a verifier as the tag it impersonates. Adversaries may interact with the tag they want to impersonate prior to, and with all other tags prior to and during the protocol run. All tags, except the impersonated tag, can be corrupted by the adversary.*

Definition 12. *Extended Soundness. Identical to Def. 11, but the adversary is given access to the* `CorruptReader` *oracle.*

E. Motivation and comparison

Our proposed model is based on the well-studied notion of (left-or-right) indistinguishability, thus avoiding the issues with less well-studied concepts such as blinders that the Vaudenay model suffers from (see Sect. III-A). Moreover, several cryptographic schemes have proven security properties based on indistinguishability games (e.g. IND-CPA, IND-CCA, IND-CCA2...). When using these schemes as building blocks for private RFID authentication protocols, it is likely to simplify proofs using our model.

Note that the Juels-Weis model from Sect. III-C also uses a traditional indistinguishability setup. However, the model requires the adversary to distinguish one out of two selected tags in the final phase. The disadvantage of this approach is that it does not take into account other properties that might leak privacy and that it limits the use of tag corruption. The Vaudenay model did introduce some crucial tools like virtual tag references and the corruption types that are still required.

Modelling details:

- The introduction of $\texttt{CreateTag}(\cdot)$: since the set of tags is not predefined we allow the adversary to dynamically create new tags.
- $\texttt{DrawTag}(\cdot,\cdot)$ and $\texttt{Free}(\cdot)_b$ are used to introduce the concept of virtual tags. This concept is needed since otherwise $\texttt{SendTag}(\cdot,\cdot)_b$ would have to accept two tag/message pairs and select one of them based on the value of the bit b. In this case it would be trivial to determine b for multi-pass protocols, simply by using different right (or left) tags for each pass of the protocol. For instance, $\texttt{SendTag}(T_0, T_1)_b$ for the first pass and $\texttt{SendTag}(T_0, T_2)_b$ for all consecutive passes. The protocol would only succeed if $b = 0$, thus allowing detection of the bit b. Hence, it is crucial that the same tag is always used within a certain protocol run, which can be ensured by using virtual tag identifiers.
- A separate communication oracle for tags and reader is used, since the reader is not considered as an entity whose privacy can be compromised.
- $\texttt{Corrupt}(\cdot)$: corruption is done with respect to a tag, not a virtual tag. If $\texttt{Corrupt}(\cdot)_b$ would accept a vtag, then determining the bit b becomes trivial by performing the following attack:
  - $vtag_i \leftarrow \texttt{DrawTag}(T_0, T_1)$
  - $C_i \leftarrow \texttt{Corrupt}(vtag_i)_b$
  - $\texttt{Free}(vtag_i)$
  - $vtag_j \leftarrow \texttt{DrawTag}(T_0, T_2)$
  - $C_j \leftarrow \texttt{Corrupt}(vtag_j)_b$

  If $C_i = C_j$ then $b = 0$, otherwise $b = 1$.
  We believe that it is realistic to assume that one has the tag identifier T_i when corrupting a tag, since corruption implies having physical access to a tag.
- The introduction of $\texttt{CreateInsider}(\cdot)$ and two new privacy notions (wide-weak-insider and wide-forward-insider): this models the case where an adversary uses the internal state a corrupted tag to attack the privacy of other tags.
- Allowing for a set of readers $\mathcal{R}$ and the introduction of $\texttt{CorruptReader}(R_j)$: with multiple readers, the probability that one will be corrupted (or is curious about the tag owner) rises. Even when a reader gets compromised, only the tag should be able to authenticate to the other readers is it registered with. Furthermore, the tag's identity should remain private when authenticating to other readers.

V. Previously Proposed Protocols

In this section, we give an overview of previously proposed protocols that are based on public key cryptography. Each of these protocols is correct, sound and achieves at least narrow-strong privacy. The reason why symmetric protocols are not considered is twofold: for private RFID symmetric authentication protocols 1) Vaudenay [9] proved that narrow-strong privacy cannot be achieved and 2) Damgård and Pedersen [17] showed that privacy comes at the cost of a non-scalable lookup procedure at the reader.

A. Zero Knowledge Based Protocols

These protocols are proofs of knowledge for a specific verifier (reader) with public key $Y = yP$. The prover (tag) proves knowledge of the private key $x \in \mathbb{Z}_\ell$, which is the discrete logarithm of the corresponding public key $X = xP$, for P a publicly agreed-on generator of $\mathbb{G}_\ell$. The public key X of the tag will serve as its identity and has been registered with the reader.

1) Randomized Schnorr: Bringer *et al.* [29] proposed Randomized Schnorr (see Fig. 2). It achieves extended soundness and narrow-strong privacy.

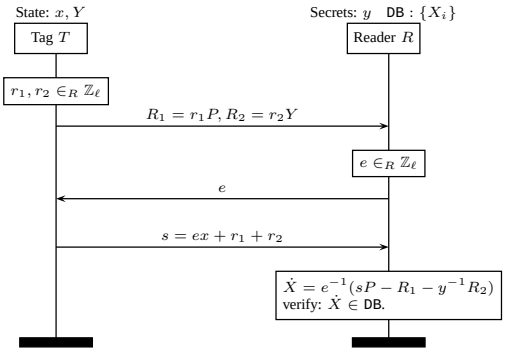


Figure 2. Randomized Schnorr [29].

2) Randomized Hashed GPS: Later, Bringer *et al.* [23] proposed Randomized Hashed GPS (see Fig. 3). The protocol has extended soundness and narrow-strong privacy. The authors also claim wide-PI-forward privacy, i.e., wide-forward privacy even when the list of registered tags' identities is known.

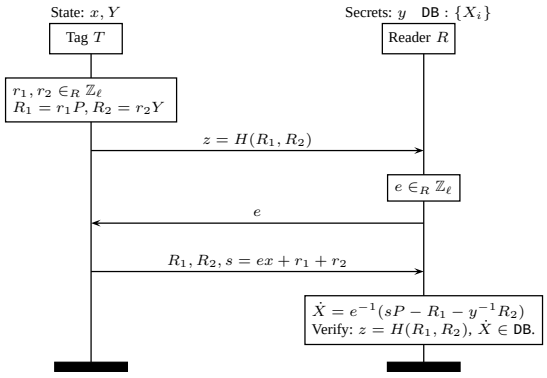


Figure 3. Randomized Hashed GPS [23].

Privacy-wise, both Randomized Schnorr and Randomized Hashed GPS suffer from the adversary having complete freedom over the exam e it sends to the tag and the fact that the final message from the tag s contains a term that is linearly dependent on this exam and the secret of the tag x. For this reason these protocols cannot be wide-strong private.[6] Furthermore, there exist a linear relation between the commitments (R_1, R_2) and the answer s. This, together with the above, makes that Randomized Schnorr cannot be wide-weak private.[7] Randomized Schnorr and Randomized Hashed GPS are also vulnerable to insider-attacks.[8]

3) IBIHOP: Peeters *et al.* [30] proposed IBIHOP (see Fig. 4). It achieves extended soundness, (reader-first) mutual authentication and wide-strong privacy.

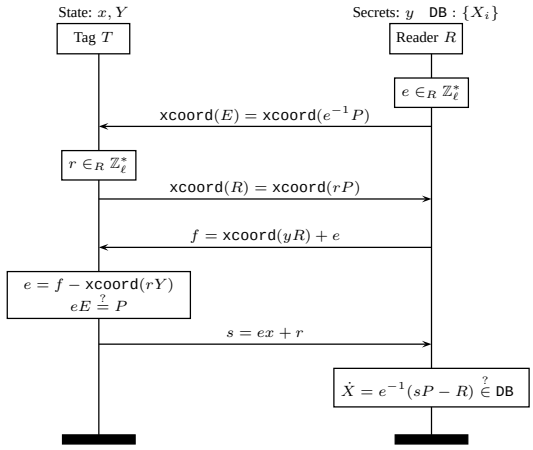


Figure 4. The IBIHOP protocol [30].

B. Public Key Encryption Based Protocols

The reader has a public/private key pair (PK, pk). The identities ID of tags that registered are stored in the reader's database. The tag and reader share a symmetric key K.

1) Vaudenay's Public Key Protocol: This protocol proposed by Vaudenay [9] (see Fig. 5) requires the tag to compute the public key encryption of one message. This cryptosystem needs to be secure against adaptive chosen ciphertext attacks (IND-CCA2) to have a secure identification scheme that achieves wide-strong privacy. One of the most efficient IND-CCA2 cryptosystems in the standard model is DHIES [13].

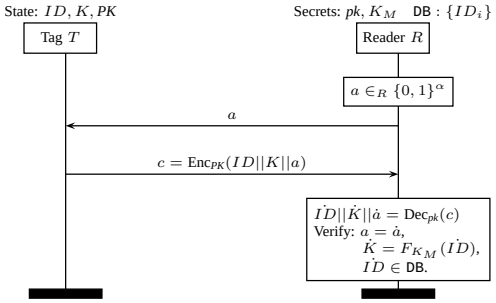


Figure 5. Vaudenay's Public Key RFID Protocol [9].

[6] An attacker in the middle sends $e - 1$ to the virtual tag and responds to the reader with $s + x$. For a correct guess of the tag's identity with known internal state x, the `result` oracle returns 1.

[7] For an observed protocol run π_0, an adversary can test, using the `result` oracle, that the current virtual tag is the tag of π_0. The adversary mounts a Man-In-The-Middle attack, sends to the reader $(R_1 + R_{1,0}, R_2 + R_{2,0})$, challenges the tag with $e - e_0$ and returns to the reader $s + s_0$.

[8] Similar to the above. The attacker sends the exam e_0 to the virtual tag in protocol run π_1. When subtracting the answers $s_0 - s_1$, the tag specific part should cancel out. The attacker starts a protocol run π_2 between its insider tag (with private key x') and the reader. The attacker sets $R_1 = R_{1,0} - R_{1,1}$, $R_2 = R_{2,0} - R_{2,1}$ and replies with $s' = s_0 - s_1 + e_2 x'$.

2) Hash ElGamal Based Protocol: Canard *et al.* [10] proposed a hash ElGamal-based protocol (see Fig. 6), which is secure, narrow-strong private and future untraceable. It is unclear how future untraceability (as defined by Canard *et al.* [10]) and wide-strong privacy relate to each other, however, these seem to be closely related. This protocol uses an IND-CPA cryptosystem, Hash ElGamal; and a MAC algorithm. It is more efficient than Vaudenay's public key scheme since the underlying encryption does not need to be IND-CCA2. Note that the combination of a MAC and IND-CPA encryption used in this specific protocol in fact provides IND-CCA2 encryption for the type of plaintext messages used [31].

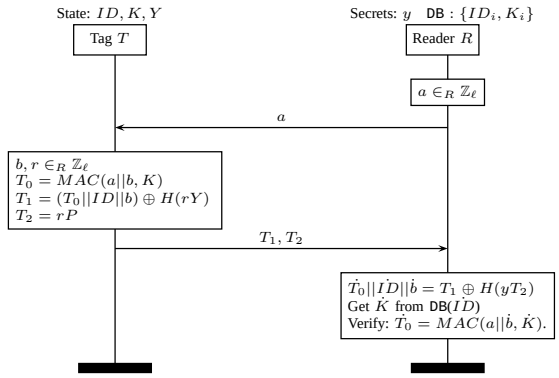


Figure 6. Hash ElGamal Based Protocol [10].

Neither protocol achieves extended soundness. Tag and reader need to store some shared (secret) data: an identifier ID and a shared secret key K. Both protocols achieve wide-strong privacy and soundness can also be proven under the more strict definition of matching conversations. Wide-strong privacy rules out insider attacks on privacy.

VI. New Wide-Forward-Insider Protocol

Our proposed protocol (see Fig. 7) is a modified version of the Schnorr [32] identification protocol. The original protocol is proven secure by Bellare and Palacio [33] under the OMDL assumption. Our starting point is a variant of the Schnorr identification protocol, where the exam of the reader is applied to the tag's randomness instead of its secret. This variant is equivalent to the original protocol, except for the case that $e = 0$. In the original Schnorr identification protocol this results in the adversary learning the tag's randomness while in the variant the adversary will learn the tag's secret. This situation is avoided by having the tag check that $e \neq 0$.

Privacy is ensured by introducing a blinding factor d that can only be computed by the tag and the reader. The blinding factor is applied to the secret x. It is important to note that the factor that is applied to the secret, only depends on input of the tag and the public key of the reader (known to the tag). As such an adversary cannot influence this value. This is an important difference with two of the previously proposed zero-knowledge based protocols (see Sect. V-A) for which the adversary can choose the factor that is applied to the secret of the tag,[9] leading to insider attacks against privacy.

[9]Due to IBIHOP's reader-first mutual authentication, an adversary cannot choose or even influence the value e.

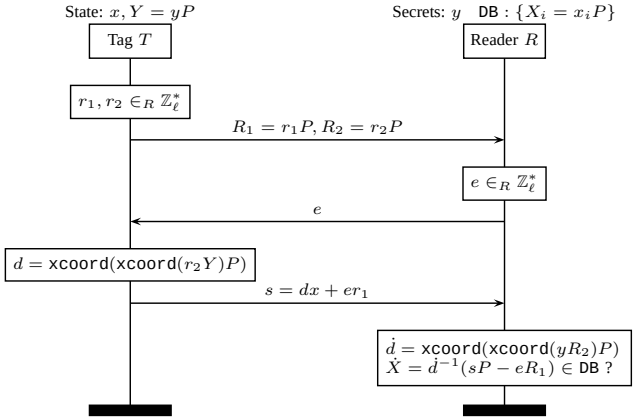


Figure 7. Wide-forward-insider private RFID identification protocol.

We will now discuss the protocol in detail. The tag generates two random numbers r_1 and r_2, where the former is needed for extended soundness and the latter is used to ensure privacy. The tag commits to its randomness by sending R_1, R_2 to the reader. The reader verifies that $R_1, R_2 \neq O$, for O the point at infinity. The tag's response is $s = dx + er_1$, with d the blinding factor as computed by the tag. Note that the tag must check that $d, e \neq 0$.[10] The reader verifies that a tag with public key $\dot{X} = \dot{d}^{-1}(sP - eR_1)$, with $\dot{d}$ the blinding factor as computed by the reader, has been registered. The reader keeps a list of all incomplete sessions. If a session timeout occurs or the tag fails to identify for a given challenge, the session is also considered to be completed.

The blinding factor contains $r_2 Y = yR_2$. Given the CDH assumption, this value can only be computed when given either r_2 or y. To prevent an adversary of exploiting the self-reducibility of the DL problem, this value is encapsulated in a one-way function. Minimising the required circuit area for implementation was one of the design criteria. Instead of using a cryptographic hash function, we choose a one-way function that only uses EC operations: $H(r_2 Y) = \mathtt{xcoord}(r_2 Y)P$. The one-wayness of this function follows directly from the DL assumption. The value d is set to the x-coordinate of the EC point.

A. Correctness

Theorem 1. *The protocol is correct according to Def. 10.*

Proof. Since $d = \mathtt{xcoord}(\mathtt{xcoord}(r_2 Y)P) = \mathtt{xcoord}(\mathtt{xcoord}(yR_2)P) = \dot{d}$, it follows that $\dot{X} = \dot{d}^{-1}(sP - eR_1) = d^{-1}((dx + er_1)P - er_1 P) = X$. □

B. Soundness

Theorem 2. *The proposed protocol has extended soundness according to Def. 11 under the OMDL assumption.*

Proof. Assume an adversary $\mathcal{A}$ that can break the extended soundness with non-negligible probability, i.e. that can perform a fresh, valid authentication with the verifier. Without

[10]By appropriate selection of the elliptic curve (e.g. a curve without points $(0, y)$), checking $d \neq 0$ is not necessary if $R_2 \neq O$.

loss of generality we will assume the target tag is known at the start of the game.[11] We construct an adversary $\mathcal{B}$ that wins the OMDL game as follows:

- Set $X = \mathcal{O}_1()$. X will be used as the public key of the target tag.
- $\mathcal{B}$ executes $\mathcal{A}$. During the first phase of $\mathcal{A}$, $\mathcal{B}$ simulates the `SendTag` oracles for the target tag as follows (all other oracles are simulated as per protocol specification):
  - On the first $\texttt{SendTag}(vtag)_b$ query of the i'th protocol run: return $R_{2,i} = r_{2,i}P$ with $r_{2,i} \in_R \mathbb{Z}_\ell$ and $R_{1,i} = \mathcal{O}_1()$.
  - On the second $\texttt{SendTag}(vtag, e_i)_b$ query of the i'th protocol run: set $d_i = \texttt{xcoord}(\texttt{xcoord}(r_{2,i}Y)P)$ and return $s_i = \mathcal{O}_2(d_iX + e_iR_{1,i})$
- During the second phase of $\mathcal{A}$, $\mathcal{B}$ proceeds as follows:
  - On the first call of $\mathcal{A}$ to $\texttt{Result}(\pi)$, compute $d = \texttt{xcoord}(\texttt{xcoord}(yR_2)P)$ and store (s, d). Next, rewind $\mathcal{A}$ until right before the call to $\texttt{SendReader}(\pi, R_1, R_2)$. On the next call to $\texttt{SendReader}(\pi, R_1, R_2)$, return a new random e'.
  - On the next call of $\mathcal{A}$ to $\texttt{Result}(\pi)$: compute $r_1 = {}^{(s-s')}/_{(e-e')}$ and $x = d^{-1}(s - er_1)$ return $(x, e_1^{-1}(s_1 - xd_1), \ldots, e_k^{-1}(s_k - xd_k))$.

The simulation by $\mathcal{B}$ is perfect during both phases. At the end of the game $\mathcal{B}$ will successfully win the OMDL with non-negligible probability, unless $s = s'$, which happens with negligible probability since both e and e' are randomly chosen after $R_2 \neq O$ is fixed. □

C. Privacy

Before giving the privacy proof we introduce a crucial conjecture that is used as building block for obtaining wide-forward-insider privacy.

Conjecture 1. *Assume a set $\mathcal{X} = \{x_1, \ldots, x_n\}$ and $\mathcal{I} = \{\iota_1, \ldots, \iota_m\}$ with $x_i, \iota_j \in_R \mathbb{Z}_\ell$ and n, m polynomial in the security parameter. We conjecture that a PPT adversary has negligible probability in winning the following game:*

1) $b \in_R \{0, 1\}$.
2) *The adversary $\mathcal{A}$ is given $\mathcal{I}$ and can interact with the system through the following oracles:*
   - $\mathcal{O}_1(\alpha, \beta) := \begin{cases} (i, d_i x_\alpha) & \text{if} \quad b = 0 \\ (i, d_i x_\beta) & \text{if} \quad b = 1 \end{cases}$ *with $d_i \in_R \mathbb{Z}_\ell$ and let i be a counter that is incremented at every call*
   - $\mathcal{O}_2(s, i) := d_i^{-1} s \in \mathcal{X} \cup \mathcal{I}$
   - $\mathcal{O}_3(s) := s \in \mathcal{X}$ [12]
3) *The adversary $\mathcal{A}$ is given $\mathcal{X}$ and outputs a bit g. The adversary wins the game if $b == g$.*

The intuition behind the experiment described above is that the adversary has a set of insider tags for which it knows the secret keys ($\mathcal{I}$) and that there is a set of tags for which the keys remains secret ($\mathcal{X}$). Through $\mathcal{O}_1$ the adversary can obtain output of the non-corrupted tags, which is a random value multiplied with the tag secret. Just as in the privacy definition, a random bit determines which tag secret x_i is selected. Since a fresh random value d_i is multiplied with every tag output, it is obvious that the adversary has negligible advantage in winning the game when only given $\mathcal{O}_1$. The oracles $\mathcal{O}_2$ and $\mathcal{O}_3$ let the adversary verify the tag output. Both oracles only return a binary value indicating whether validation succeeded. The random d_i's are used in $\mathcal{O}_2$ to verify the input. Intuitively, the only way that the adversary can win the game is by either guessing some x_i and checking it through oracle $\mathcal{O}_3$ or by giving an input (s, i) to $\mathcal{O}_2$ that did not directly originate from a call to $\mathcal{O}_1$ (i.e. that maps to a different x_i than the call to $\mathcal{O}_1$ did). The probability of both these events happening however seems negligible.

Theorem 3. *The proposed protocol is wide-forward-insider and narrow-strong private according to Def. 9 under the ODH assumption, the XL assumption and Conjecture 1.*

Proof. Assume an adversary $\mathcal{A}$ that wins the privacy game with non-negligible advantage. Using a standard hybrid argument [34], [35], we construct an adversary that breaks the ODH-assumption. We set $Y = B$. $\mathcal{B}_i$ plays the privacy game with $\mathcal{A}$. $\mathcal{B}_i$ selects a random bit $\tilde{b}$, which will indicate which world is simulated to $\mathcal{A}$. All oracles are simulated in the regular way, with the exception of the `SendTag` and `Result` oracle for the target tag:

- $\texttt{SendTag}(vtag)_b$:
  - $j \neq i$: Generate $r_1, r_2 \in_R \mathbb{Z}_\ell$. Take $R_1 = r_1P, R_2 = r_2P$. Return R_1 and R_2.
  - $j = i$: Generate $r_1 \in_R \mathbb{Z}_\ell$. Take $R_1 = r_1P, R_2 = A$. Return R_1 and R_2.
- $\texttt{SendTag}(vtag, e)_b$, j'th query: retrieve the tuple $(vtag, T_0, T_1)$ from the table $\mathcal{D}$. Take the key x for tag $T_{\tilde{b}}$.
  - $j < i$: Generate $r \in_R \mathbb{Z}_\ell$. Take $d = \texttt{xcoord}(H(rP))$. Return $s = dx + er_1$.
  - $j = i$: Take $d = \texttt{xcoord}(H(C))$. Return $s = dx + er_1$.
  - $j > i$: Take $d = \texttt{xcoord}(H(r_2Y))$. Return $s = dx + er_1$.
- $\texttt{Result}(\pi)$: If the received R_2 in session π matches A from the ODH problem take $\dot{d} = \texttt{xcoord}(H(C))$. If not, check if R_2 matches any of the R_2's generated during the first $i - 1$ `SendTag` queries. If so, use the r generated in that query and compute $\dot{d} = \texttt{xcoord}(H(rP))$. Otherwise, take $\dot{d} = \texttt{xcoord}(\mathcal{O}(R_2))$. Finally, compute $\dot{X} = \dot{d}^{-1}(sP - eR_1)$. Check $\dot{X}$ with the database, return true if $\dot{X}$ is found, false otherwise.

At the end of the game $\mathcal{A}$ outputs its guess g for the privacy game. $\mathcal{B}_i$ outputs $(\tilde{b} == g)$.

[11] Otherwise, the proof can be adapted by choosing the public keys of the tags as $X_i = \mathcal{O}_1()$. All tag queries are simulated as for the target tag, until the tag is corrupted. When corrupting a tag, call $\mathcal{O}_2(X_i)$ for that tag and use the result as private key for simulating all following queries to that tag. At the end of the game, use the $\mathcal{O}_2(\cdot)$ oracle to extract all remaining discrete logarithms, except for the target tag.

[12] Due to a technicality in the privacy proof, we need to replace this oracle by $\mathcal{O}_3(S) := dlog(S) \in \mathcal{X}$. Note that it is the challenger, which is computationally unbounded, that computes the discrete logarithm in this oracle. This definition is equivalent to the one given here, since the adversary can always call $\mathcal{O}_3$ with sP instead of s.

The above simulation to $\mathcal{A}$ is perfect, since validation is done in the same way as the protocol specification. If $R_2 = A$, the oracle $\mathcal{O}(\cdot)$ cannot be used. However, in this case we know the corresponding value of d by directly using $H(C)$, which gives the same result.

We use $\mathcal{A}^i$ (with $i \in [1 \dots k]$) to denote the case that $\mathcal{A}$ runs with the first i `SendTag` queries random instances, and the other queries real instances. This is the case when $\mathcal{B}_{i+1}$ runs with a real ODH instance, or $\mathcal{B}_i$ with a random ODH instance. Note that $\mathcal{A}^i$ wins if $\tilde{b} == g$.

By the hybrid argument we get that:

$$\| \Pr\left[\mathcal{A}^0 \text{ wins}\right] - \Pr\left[\mathcal{A}^k \text{ wins}\right] \| \leq \sum \mathbf{Adv}_{B_i} .$$

- In the case of $\mathcal{A}^0$, it is clear $\Pr\left[\mathcal{A}^0 \text{ wins}\right] = \Pr\left[\mathcal{A} \text{ wins}\right]$ since all oracles are simulated exactly as in the protocol definition.
- In the case of $\mathcal{A}^k$, all `SendTag` queries are simulated with $r \in_R \mathbb{Z}_\ell$ and $d = \texttt{xcoord}(\texttt{xcoord}(rP)P)$. Under the XL assumption it follows that d is indistinguishable from a random value from the x-coordinate distribution and that d is independent of R_1, R_2 and e.

a) Narrow-strong privacy: Since $s = dx + er_1$ and $R_1 = r_1 P$, it follows under the XL assumption that (s, e, R_1), with d a random value from the x-coordinate distribution, is indistinguishable from $(\tilde{r}, e, R)$, with $\tilde{r}$ a uniformly random value. Hence it follows that s is indistinguishable from a uniformly random value independent of x, as long as $e, d \neq 0$. Note that this only holds in the absence of a `Result` oracle (which is able to distinguish $\tilde{r}$ from random).

So $\mathcal{A}^k$ has probability $1/2$ of winning the privacy game, since it obtains no information at all on x from a tag.

$$\begin{aligned} \| \Pr\left[\mathcal{A}^0 \text{ wins}\right] - \Pr\left[\mathcal{A}^k \text{ wins}\right] \| &= \| \Pr\left[\mathcal{A} \text{ wins}\right] - \frac{1}{2} \| \\ &= \frac{1}{2}\mathbf{Adv}^{privacy}_{\mathcal{A}} \\ &\leq \sum \mathbf{Adv}_{B_i} \end{aligned}$$

It follows that at least one of the $\mathcal{B}_i$ has non-negligible probability to win the ODH game.

b) Wide-forward-insider privacy: For proving wide-forward-insider privacy, we also have to simulate the `Result` oracle, which was ommitted in the case of narrow-strong privacy. We can now do a straightforward reduction to the game from Conjecture 1. All $\texttt{SendTag}(vtag, e)_b$ calls are simulated using $\mathcal{O}_1(i, j)$ for the tags T_i and T_j passed to `DrawTag`. Calls to `Result` are simulated using $\mathcal{O}_2(sP - eR_1, i)$ if the R_1 received by the server matches an R_1 resulting from a $\texttt{SendTag}(vtag)_b$, otherwise $\dot{d}$ is computed as in the original protocol and $\mathcal{O}_3(\dot{d}^{-1}(sP - eR_1))$ is used to validate the resulting secret. □

VII. Efficiency Optimisation

The protocol will be optimised by reducing the computational effort at the tag side. This is done by having the tag only generate one random value r ($r_1 = r_2$). As such, the tag has to compute one less scalar-EC point multiplication and has to transmit one less element. We also change the blinding factor to $d = \texttt{xcoord}(rY)$. This reduces the computational effort required from the tag with another scalar-EC point multiplication. An overview of the protocol is given in Fig. 8.

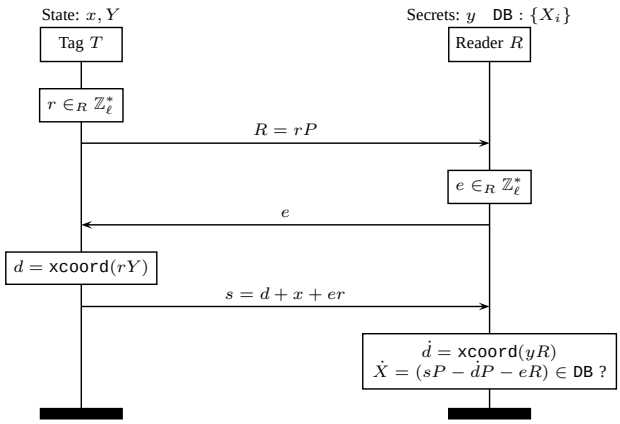


Figure 8. Optimised protocol.

A. Soundness

Theorem 4. *The optimised protocol has extended soundness according to Def. 12 under the OMDL assumption.*

The proof is similar to the one of the original protocol.

B. Privacy

For the privacy proofs, an extended ODH variant is required. To be able to verify the tag's response, we only need $dP = \texttt{xcoord}(rY)P$, which is already available in the original ODH by oracle $\mathcal{O}(Z) = H(bZ)$. However, to also be able to compute s, in the case that $R = A$, the value $\texttt{xcoord}(C)+ea$ is required. In our extended ODH variant, the adversary has access to the oracle $\mathcal{O}'(z) := \texttt{xcoord}(C) + za$ that can be called once with $z \neq 0$, in addition to $A = aP, B = bP, \texttt{xcoord}(C)P$ and the oracle $\mathcal{O}(Z)$ of the original ODH.

Theorem 5. *The optimised protocol is narrow-strong and wide-forward-insider private according to Def. 9 under our extended ODH assumption and the additive variant of Conj.1.*

Proof. Assume an adversary $\mathcal{A}$ that wins the privacy game with non-negligible advantage. Using a hybrid argument we construct an adversary that breaks the ODH-assumption. We set $Y = B$. $\mathcal{B}_i$ plays the privacy game with $\mathcal{A}$. $\mathcal{B}_i$ selects a random bit $\tilde{b}$, which will indicate which world is simulated to $\mathcal{A}$. All oracles are simulated in the regular way, with the exception of the `SendTag` and `Result` oracle for the target tag:

- $\texttt{SendTag}(vtag)_b$:
  - $j \neq i$: Generate $r \in_R \mathbb{Z}_\ell$. Take $R = rP$. Return R.
  - $j = i$: Take $R = A$. Return R.
- $\texttt{SendTag}(vtag, e)_b$, j'th query: retrieve the tuple $(vtag, T_0, T_1)$ from the table $\mathcal{D}$. Take the key x for tag $T_{\tilde{b}}$.
  - $j < i$: Generate $r' \in_R \mathbb{Z}_\ell$. Take $d = \texttt{xcoord}(r'P)$. Return $s = x + er + d$.
  - $j = i$: Return $s = x + \mathcal{O}'(e)$.

  - $j > i$: Take $d = \texttt{xcoord}(rY)$. Return $s = x + er + d$.

- $\texttt{Result}(\pi)$: If the received R in session π matches A from the ODH problem take $\dot{d}P = \texttt{xcoord}(C)P$. If not, check if R matches any of the R's generated during the first $i-1$ SendTag queries. If so, use the r' generated in that query and compute $\dot{d}P = \texttt{xcoord}(r'P)P$. Otherwise, take $\dot{d}P = \mathcal{O}(R)$. Finally, compute $\dot{X} = sP - eR - \dot{d}P$. Check $\dot{X}$ with the database, return true if $\dot{X}$ is found, false otherwise.

At the end of the game $\mathcal{A}$ outputs its guess g for the privacy game. $\mathcal{B}_i$ outputs $(\tilde{b} == g)$.

The above simulation to $\mathcal{A}$ is perfect, since validation is done in the same way as the protocol specification. If $R = A$, the oracle $\mathcal{O}(\cdot)$ cannot be used. However, in this case we know the corresponding value of dP by directly using $\texttt{xcoord}(C)P$, which gives the same result.

We use $\mathcal{A}^i$ (with $i \in [1 \dots k]$) to denote the case that $\mathcal{A}$ runs with the first i SendTag queries random instances, and the other queries real instances. This is the case when $\mathcal{B}_{i+1}$ runs with a real ODH instance, or $\mathcal{B}_i$ with a random ODH instance. Note that $\mathcal{A}^i$ wins if $\tilde{b} == g$.

By the hybrid argument we get that:

$$\| \Pr\left[\mathcal{A}^0 \text{ wins}\right] - \Pr\left[\mathcal{A}^k \text{ wins}\right] \| \leq \sum \mathbf{Adv}_{B_i} .$$

- In the case of $\mathcal{A}^0$, it is clear $\Pr\left[\mathcal{A}^0 \text{ wins}\right] = \Pr\left[\mathcal{A} \text{ wins}\right]$ since all oracles are simulated exactly as in the protocol definition.
- In the case of $\mathcal{A}^k$, all SendTag queries are simulated with $r \in_R \mathbb{Z}_\ell$ and $d = \texttt{xcoord}(rP)$.

a) Narrow-strong privacy: Since $s = x + er + d$ and $R = rP$, it follows under the XL assumption that (s, e, R), with d a random value from the x-coordinate distribution, is indistinguishable from $(\tilde{r}, e, R)$, with $\tilde{r}$ a uniformly random value. Hence it follows that s is indistinguishable from a uniformly random value independent of x, as long as $e, d \neq 0$.

So $\mathcal{A}^k$ has probability $1/2$ of winning the privacy game, since it obtains no information at all on x from a tag.

$$\begin{aligned} \| \Pr\left[\mathcal{A}^0 \text{ wins}\right] - \Pr\left[\mathcal{A}^k \text{ wins}\right] \| &= \| \Pr\left[\mathcal{A} \text{ wins}\right] - \frac{1}{2} \| \\ &= \frac{1}{2}\mathbf{Adv}_{\mathcal{A}}^{privacy} \\ &\leq \sum \mathbf{Adv}_{B_i} \end{aligned}$$

It follows that at least one of the $\mathcal{B}_i$ has non-negligible probability to win the ODH game.

b) Wide-forward-insider privacy: For proving wide-forward-insider privacy, we also have to simulate the Result oracle, which was ommitted in the case of narrow-strong privacy. We can now do a straightforward reduction to the game from (the additive variant of) Conjecture 1. All $\texttt{SendTag}(vtag, e)_b$ calls are simulated using $\mathcal{O}_1(i, j)$ for the tags T_i and T_j passed to DrawTag. Calls to Result are simulated using $\mathcal{O}_2()$ if the R_1 received by the server matches an R_1 resulting from a $\texttt{SendTag}(vtag)_b$, otherwise $\dot{d}$ is computed as in the original protocol and $\mathcal{O}_3()$ is used to validate the resulting secret. □

VIII. Implementation Considerations

Our protocol mainly requires the evaluation of scalar-EC point multiplications and the generation of a random number. For 80 bit security, we need an elliptic curve over a field that is approximately 160 bits in size. The protocol can be implemented on the architecture proposed by Lee *et al.* [36]. Their ECC coprocessor can be built with less than 15 kGEs (Gate Equivalent), consumes $\pm 13, 8\mu W$ of power and takes around 85 ms for one scalar-EC point multiplication. More recently, Wenger and Hutter[37] proposed an ECC coprocessor that only requires 9 kGEs, consumes $\pm 32, 3\mu W$ of power and takes around 286 ms for one scalar-EC point multiplication. Aside from the ECC coprocessor, circuit area is required for the ROM (Read Only Memory), RAM (Random Access Memory) and RNG (Random Number Generator).

A. (Non-)Sense of Coupons

Several papers have proposed to use precomputation as an optimisation. The protocol is split in an off-line and on-line phase, for which the on-line phase is more efficient (faster) than the original protocol. The precomputed values are stored in the form of coupons. There are two options: either these coupons are precomputed externally and pushed on the tag or the tag generates these coupons itself.

Computing coupons externally has the additional benefit that for most protocols, less logic needs to be implemented on the tag. The downside of the tag itself not being able to do these necessary computations is that an adversary can quite easily mount a denial of service attack, by tricking the tag into authenticating over and over until it has no coupons left. This attack could be prevented by introducing mutual authentication, more specifically have the reader first authenticate to the tag. Ironically the only efficient way to achieve authentication of the reader to a yet-unknown tag, is by using zero knowledge proofs, which in turn require a full fletched ECC coprocessor on the tag to verify these. How to securely push these coupons onto the tag is an additional issue.

Having the tag itself precompute coupons can speed up the identification process, or alternatively make it possible to use a slower EC coprocessor that is smaller. The tag computes these coupons whenever energy is available. A tag can draw energy as long as it is in the electromagnetic field around any reader. Since the tag can do all the necessary computations itself, one only needs limited storage for b coupons.

The disadvantage of coupons is that these need to be stored on the tag. When making abstraction of the control logic needed to access this storage, one still needs about one floating gate per bit. The size of the coupons can be minimized by not storing the used randomness but instead implementing a pseudo-random function with a seed to generate random numbers on the tag. Taking this optimisation into account, the protocols discussed in this paper still require coupons that consist of two EC points. For an area of 10 KGe (≈ the size of Wenger and Hutter's ECC coprocessor) one can store 20 to 30 of such coupons.

In general it can be argued that strong privacy is not achievable when using coupons or a pseudo-random function

instead of a true random number generator. By making a query to the `Corrupt` oracle, the adversary learns the complete internal state of the tag, which also comprises coupons and/or the seed of the pseudo-random function. When the coupons are generated by the tag itself, b-strong privacy is possible, meaning that the tag is unlinkable again after b conversations from the moment the tag was freed.

At least part of the coupons is reader specific. This puts an additional burden on tag storage requirements in the multi-reader setting, making the use of coupons impractical.

B. Comparison

Table I gives a comparision of our protocol to previously proposed protocols, described in Sect. V.

Both the Randomized Schnorr and our proposed protocol benefit from a compact hardware design: only an ECC co-processor is needed. The other protocols require additional hardware to evaluate a cryptographic hash function, which makes the design substantially larger. Current cryptographic hash functions [38] require at least 50% of the circuit area of the most compact ECC implementation.

The scalar-EC point multiplication is more complex than the evaluation of a hash/MAC. For a fair comparison between the performance of protocols that require the evaluation of a hash/MAC and protocols that do not, one should assume the same total available circuit size. This means that our protocol can be implemented using a larger but faster ECC processor.

When also considering the more general setting, where a single tag can identify the end-user privately to multiple readers, the tags not only need to store an extra public key for every reader but also corresponding shared data, if any. In this setting there is a clear advantage for protocols that provide extended soundness, since the tag can use the same private/public key pair to identify to each reader.

IX. Conclusion

RFID privacy was approached from both the modelling and protocol point of view. Several RFID privacy models were critically examined with respect to their assumptions, practical usability and other issues that arise when applying their privacy definition to concrete protocols. Some models are based on unrealistic assumptions, others are impractical to apply. We presented a new RFID privacy model, based on the classic notion of indistinguishability, that combines the benefits of existing models while avoiding their identified drawbacks. Since this privacy model is based on an indistinguishability game, one can rely on a wide range of existing proof techniques, making the model quite straightforward to use in practice. Furthermore, this is the first model that allows for a more general setting where a tag can privately authenticate to multiple (independent) readers. This model also incorporates the creation of insider tags, in order to also capture privacy attacks for which the corrupted tag is not the one under attack. We showed that the combination of the notions narrow-strong and wide-forward-insider is sufficient for most practical applications.

From the protocol side, we examined several protocols towards their security and privacy properties. We proposed a new wide-forward-insider and narrow-strong private zero-knowledge RFID identification protocol and its optimised version. Security and privacy of our proposed protocols are proven in the standard model. Our protocol is the most efficient privacy-preserving RFID authentication protocol for most practical applications and can be implemented on the tags, using only Elliptic Curve Cryptography. This allows for a compact hardware design and requires minimal computational effort from the tag, namely two scalar-EC point multiplications. As an additional benefit, our protocols do not require any shared secrets between readers and tags. This makes these protocols very suitable for use with multiple readers.

Acknowledgements

The authors would like to thank everyone that contributed to some very fruitful discussions, came up with possible proof strategies, provided useful suggestions or tried to break the claimed privacy properties of the proposed protocols. Special thanks to: Julien Bringer, Ivan Damgård, Junfeng Fan, Jesper Buus Nielsen, Andreas Pashalidis, Dominique Raub, Koen Simoens, Dave Singelée, Serge Vaudenay and Frederik Vercauteren. The work leading to these results has received funding from the European Community's Framework Programme (FP7/2007-2013) under grant agreement n^o 284862, and the Research Council KU Leuven: GOA TENSE (GOA/11/007).

References

[1] S. A. Weis, S. E. Sarma, R. L. Rivest, and D. W. Engels, "Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems," in *SPC*, ser. LNCS, vol. 2802. Springer, 2003, pp. 201–212.

[2] S. Mangard, E. Oswald, and T. Popp, *Power analysis attacks - revealing the secrets of smart cards*. Springer, 2007.

[3] M. Hutter, J.-M. Schmidt, and T. Plos, "RFID and Its Vulnerability to Faults," in *CHES*, ser. LNCS, vol. 5154. Springer, 2008, pp. 363–379.

[4] T. Kasper, D. Oswald, and C. Paar, "New Methods for Cost-Effective Side-Channel Attacks on Cryptographic RFIDs," *RFIDSec*, 2009.

[5] R. Canetti, O. Goldreich, S. Goldwasser, and S. Micali, "Resettable zero-knowledge (extended abstract)," in *STOC*, 2000, pp. 235–244.

[6] M. Bellare, M. Fischlin, S. Goldwasser, and S. Micali, "Identification Protocols Secure against Reset Attacks," in *EUROCRYPT*, ser. LNCS, vol. 2045. Springer, 2001, pp. 495–511.

[7] V. Goyal and A. Sahai, "Resettably Secure Computation," in *EUROCRYPT*, ser. LNCS, vol. 5479. Springer, 2009, pp. 54–71.

[8] T. van Deursen and S. Radomirović, "Insider attacks and privacy of RFID protocols," in *EuroPKI*, ser. LNCS, vol. 7163. Springer, 2011, pp. 65–80.

[9] S. Vaudenay, "On Privacy Models for RFID," in *ASIACRYPT*, ser. LNCS, vol. 4833. Springer, 2007, pp. 68–87.

[10] S. Canard, I. Coisel, J. Etrog, and M. Girault, "Privacy-Preserving RFID Systems: Model and Constructions," Cryptology ePrint Archive, Report 2010/405, 2010, http://eprint.iacr.org/.

[11] M. Bellare, C. Namprempre, D. Pointcheval, and M. Semanko, "The One-More-RSA-Inversion Problems and the Security of Chaum's Blind Signature Scheme," *Journal of Cryptology*, vol. 16, pp. 185–215, 2003.

[12] D. R. L. Brown and K. Gjøsteen, "A Security Analysis of the NIST SP 800-90 Elliptic Curve Random Number Generator," in *CRYPTO*, ser. LNCS, vol. 4622. Springer, 2007, pp. 466–481.

[13] M. Abdalla, M. Bellare, and P. Rogaway, "The Oracle Diffie-Hellman Assumptions and an Analysis of DHIES," in *CT-RSA*, ser. LNCS, vol. 2020. Springer, 2001, pp. 143–158.

[14] G. Avoine, E. Dysli, and P. Oechslin, "Reducing Time Complexity in RFID Systems," in *SAC*, ser. LNCS, vol. 3897. Springer, 2005, pp. 291–306.

[15] M. Burmester, T. Le, and B. Medeiros, "Provably secure ubiquitous systems: Universally composable RFID authentication protocols," in *SECURECOMM*. IEEE Press, 2006.

[16] T. Van Le, M. Burmester, and B. de Medeiros, "Universally composable and forward-secure RFID authentication and authenticated key exchange," in *ASIACCS*. ACM, 2007, pp. 242–252.

Table I
OVERVIEW OF DIFFERENT PROPOSED PROTOCOLS.

Protocol	Strongest Privacy	Insider Private	Extended Soundness	Mutual	Tag Operations	Reader Operations
Randomized Schnorr [29]	narrow-strong	no	yes	no	2 EC mult	3 EC mult 3 EC add
Randomized Hashed GPS [23]	narrow-strong wide-forward	no	yes	no	2 EC mult 1 hash	3 EC mult 3 EC add 1 hash
IBIHOP [30]	wide-strong	yes	yes	yes	3 EC mult	4 EC mult 1 EC add
Vaudenay [9] + DHIES [13]	wide-strong	yes	no	no	2 EC mult 2 hash/MAC 1 symm enc	1 EC mult 3 hash/MAC 1 symm dec
Hash ElGamal [10]	wide-strong	yes	no	no	2 EC mult 2 hash/MAC	1 EC mult 2 hash/MAC
Proposed Protocol (Sect. VI)	wide-forward-insider	yes	yes	no	4 EC mult	4 EC mult 1 EC add
Optimised Protocol (Sect. VII)	wide-forward-insider	yes	yes	no	2 EC mult	3 EC mult 2 EC add

[17] I. Damgård and M. Ø. Pedersen, "RFID Security: Tradeoffs between Security and Efficiency," in *CT-RSA*, ser. LNCS, vol. 4964. Springer, 2008, pp. 318–332.

[18] J. Ha, S.-J. Moon, J. Zhou, and J. Ha, "A New Formal Proof Model for RFID Location Privacy," in *ESORICS*, vol. LNCS. Springer, 2008, pp. 267–281.

[19] A. Juels and S. A. Weis, "Defining Strong Privacy for RFID," in *PerCom Workshops*. IEEE Computer Society, 2007, pp. 342–347.

[20] R.-I. Paise and S. Vaudenay, "Mutual Authentication in RFID: Security and Privacy," in *ASIACCS*. ACM Press, 2008, pp. 292–299.

[21] C. Y. Ng, W. Susilo, Y. Mu, and R. Safavi-Naini, "RFID Privacy Models Revisited," in *ESORICS*, ser. LNCS, vol. 5283. Springer, 2008, pp. 251–266.

[22] ——, "New Privacy Results on Synchronized RFID Authentication Protocols against Tag Tracing," in *ESORICS*, ser. LNCS, vol. 5789. Springer, 2009, pp. 321–336.

[23] J. Bringer, H. Chabanne, and T. Icart, "Efficient Zero-Knowledge Identification Schemes which respect Privacy," in *ASIACCS*. ACM, 2009, pp. 195–205.

[24] A.-R. Sadeghi, I. Visconti, and C. Wachsmann, "Anonymizer-Enabled Security and Privacy for RFID," in *CANS*, ser. LNCS, vol. 5888. Springer, 2009, pp. 134–153.

[25] F. Armknecht, A.-R. Sadeghi, A. Scafuro, I. Visconti, and C. Wachsmann, "Impossibility Results for RFID Privacy Notions," in *Transactions on Computational Science IX*. Springer, 2010, pp. 39–63.

[26] K. Ouafi and S. Vaudenay, "Strong Privacy for RFID Systems from Plaintext-Aware Encryption," in *CANS*, ser. LNCS, vol. 7712. Springer, 2012, pp. 247–262.

[27] D. Bleichenbacher, "Chosen Ciphertext Attacks Against Protocols Based on the RSA Encryption Standard PKCS #1," in *CRYPTO*, ser. LNCS, vol. 1462. Springer, 1998, pp. 1–12.

[28] J.-M. Bohli and A. Pashalidis, "Relations Among Privacy Notions," in *Financial Cryptography*, ser. LNCS, vol. 5628. Springer, 2009, pp. 362–380.

[29] J. Bringer, H. Chabanne, and T. Icart, "Cryptanalysis of EC-RAC, a RFID Identification Protocol," in *CANS*, ser. LNCS, vol. 5339. Springer-Verlag, 2008, pp. 149–161.

[30] R. Peeters, J. Hermans, and J. Fan, "IBIHOP: Proper Privacy Preserving Mutual RFID Authentication," in *RFIDSec Asia 2013*, ser. Cryptology and Information Security, vol. 11. IOS PRESS, 2013, pp. 45–56.

[31] H. Krawczyk, "The Order of Encryption and Authentication for Protecting Communications (or: How Secure Is SSL?)," in *CRYPTO*, ser. LNCS, vol. 2139. Springer, 2001, pp. 310–331.

[32] C.-P. Schnorr, "Efficient Signature Generation by Smart Cards," *J. Cryptology*, vol. 4, no. 3, pp. 161–174, 1991.

[33] M. Bellare and A. Palacio, "GQ and Schnorr Identification Schemes: Proofs of Security against Impersonation under Active and Concurrent Attacks," in *CRYPTO*, ser. LNCS, vol. 2442. Springer, 2002, pp. 162–177.

[34] A. C.-C. Yao, "Theory and applications of trapdoor functions (extended abstract)," in *FOCS*. IEEE Computer Society, 1982, pp. 80–91.

[35] O. Goldreich, *Foundations of Cryptography: Volume 1, Basic Tools*. Cambridge University Press, 2001.

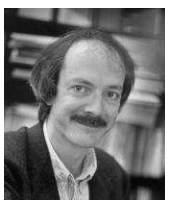
[36] Y. K. Lee, L. Batina, D. Singelée, and I. Verbauwhede, "Low-Cost Untraceable Authentication Protocols for RFID," in *WiSec*. ACM, 2010, pp. 55–64.

[37] E. Wenger and M. Hutter, "A Hardware Processor Supporting Elliptic Curve Cryptography for Less Than 9 kGEs," in *CARDIS*, ser. LNCS, vol. 7079. Springer, 2011, pp. 182–198.

[38] SHA-3 Zoo. Overview of all candidates for the SHA-3 hash competition organized by NIST. http://ehash.iaik.tugraz.at/wiki/The_SHA-3_Zoo.

Jens Hermans is a postdoctoral researcher at the research group COSIC at KU Leuven (Belgium). He obtained a Master's degree in Mathematical Engineering and a Phd in Engineering Sciences at KU Leuven in 2008 and 2012 respectively. His research interest are the analysis and design of cryptographic protocols, provable security and implementation of public key cryptography.

Roel Peeters is a postdoctoral researcher at the research group COSIC at KU Leuven. He obtained a Master's degree in Electrical Engineering and a Phd in Engineering Sciences at KU Leuven in 2007 and 2012 respectively. His research interest are mainly in the analysis and design of cryptographic protocols. He authored more than 20 scientific publications, and participated in various research projects.



Bart Preneel is a full professor at the KU Leuven where he heads the COSIC research group. He received the Electrical Engineering degree and the Doctorate in Applied Sciences from KU Leuven. He was visiting professor at five universities in Europe and was a research fellow at the University of California at Berkeley. Bart Preneel coordinated the EU Network of Excellence ECRYPT and served as panel member and chair for the European Research Council. Since 1997 he is serving on the Board of Directors of the IACR (International Association for Cryptologic Research), from 2002-2007 as vice president and from 2008-2013 as president. He was program chair of 15 international conferences and he has been invited speaker at more than 90 conferences in 40 countries. His main research interests are cryptography, information security and privacy. He authored more than 400 scientific publications and is inventor of 4 patents.